CLAIMS

What is claimed is:

1. A method of performing a peer-to-peer DMA to efficiently move data between a first DMA capable ASIC block and a second DMA capable ASIC block over an AHB bus architecture, said method comprising:

initiating a DMA by setting up a first ASIC block to access a second ASIC block;

holding off the DMA transfer until the second ASIC block is set up and ready for the transfer; and

transferring data between the second ASIC block and the first ASIC block after the second ASIC block is set up and ready for the DMA transfer.

2. A method according to claim 1, wherein the first ASIC block initiates the DMA by accessing a specific address that corresponds to the second ASIC block.

3. A method according to claim 1, wherein a master interface of the first ASIC block is used to initiate the DMA transfer and to generate control signals and data signals for the second ASIC block, and wherein the control signals and data for the second ASIC block are routed through a slave interface of the second ASIC block.

4. A method according to claim 3, wherein signals of the second ASIC block are mapped from a master interface of the second ASIC block to the slave interface of the second ASIC block through muxes.

5. A method according to claim 1, wherein the second ASIC block is set up as if it is going to perform a DMA in the opposite direction as the first ASIC block.

6. A method according to claim 1, wherein holding off the DMA transfer comprises sending a split response until the second ASIC block is set up and ready to perform the DMA transfer.

7. A method according to claim 1, wherein the DMA transfer takes place without accessing system memory.

8. A method according to claim 1, wherein the DMA transfer is performed without the use of sideband signals.

9. A method according to claim 1, wherein the DMA transfer is performed without duplicate hardware for interfacing core blocks on another side of the ASIC blocks.

10. A peer-to-peer capable AHB bus architecture, comprising:

a first DMA capable ASIC block comprising an AHB master interface;

a second DMA capable ASIC block comprising an AHB master interface, wherein the first ASIC block is configured to initiate a peer-to-peer DMA with the second ASIC block;

an AHB slave interface;

a mux disposed between the AHB master interface of the second ASIC block and its AHB slave interface; and

a bus arbiter configured to grant access to the bus by the ASIC blocks.

11. A bus architecture according to claim 10, wherein each of the ASIC blocks comprises both a master interface and a slave interface.

12. A bus architecture according to claim 10, further comprising a plurality of muxes including a write data mux and a read data mux.

13. A bus architecture according to claim 10, wherein the bus arbiter is configured to grant bus access to only one ASIC block.

14. A bus architecture according to claim 10, wherein the second ASIC block comprises:

internal logic configured to determine when the second ASIC block is being accessed by the first ASIC block;

a first mux configured to disconnect a bus request signal from the arbiter when the second ASIC block is being accessed by the first ASIC block;

a second mux configured to disconnect a bus grant signal from the arbiter when the second ASIC block is being accessed by the first ASIC block; and

a slave-to-slave bridge configured to internally generate the bus grant signal to cause the second ASIC block to operate as if it had bus access from the arbiter, and further configured to use the bus request signal to determine when the second ASIC block is set up and ready for the DMA transfer.

15. A peer-to-peer capable ASIC block, comprising:

internal logic configured to determine when the ASIC block is being accessed by another ASIC block;

a first mux configured to disconnect a bus request signal from the arbiter when the ASIC block is being accessed by another ASIC block;

a second mux configured to disconnect a bus grant signal from the arbiter when the ASIC block is being accessed by another ASIC block; and

a slave-to-slave bridge configured to internally generate the bus grant signal to cause the ASIC block to operate as if it had bus access granted from an arbiter, and further configured to use the bus request signal to determine when the ASIC block is set up and ready for a DMA transfer.

16. A peer-to-peer capable ASIC block according to claim 15, further comprising a DMA signal comparator.

17. A peer-to-peer capable ASIC block according to claim 15, further comprising:

a source/destination register;

a burst size register; and

a DMA control register.

18. A peer-to-peer capable ASIC block according to claim 15, further comprising:
a first read data mux;
a second read data mux; and
a first write data mux.

19. A peer-to-peer capable ASIC block according to claim 15, wherein the slave-to-slave bridge includes an address decoder.

20. A peer-to-peer capable ASIC block according to claim 15 wherein the slave-to-slave bridge is configured to map the master interface control signals to slave control signals.

21. A method of establishing a peer-to-peer DMA, comprising:
initiating a peer-to-peer DMA from a first DMA capable ASIC block;
setting up a second DMA capable ASIC block to communicate with the initiating ASIC block; and
routing control and data signals between the second ASIC block and the first ASIC block.

22. A method according to claim 21, wherein setting up the second DMA capable ASIC block to communicate with the initiating ASIC block comprises configuring the second ASIC block as if it is going to perform a normal DMA in an opposite direction as the initiating ASIC block.

23. A method according to claim 21, wherein routing the control and data signals comprises routing the control and data signals through a slave interface using muxes.

24. A method according to claim 21, wherein each of the DMA capable ASIC blocks comprises a master interface configured to initiate reads and writes to the system memory and a slave interface configured to read and write registers that setup and run the DMA.

25. A method according to claim 21, wherein the second DMA capable ASIC block uses a split response to pause the DMA in the initiating ASIC block until the second block is ready to begin the transfer.

26. A method according to claim 21, wherein the initiating DMA capable ASIC block is set up as if it is going to read or write to a specific address.

27. A method of transferring data from a first DMA capable ASIC block to a second DMA capable ASIC block across a bus, the method comprising:

mapping a slave interface of the first block to a master interface of the second block; and

initiating a transfer of the data at the first logic block while holding off the transfer of the data at the second block until the second block is set up and ready to complete the transfer.

28. A method according to claim 27, wherein the transfer of data is held off by sending a split response until the second block is set up and ready to complete the transfer.

29. A method according to claim 27, wherein both the first block and the second block are set up as if they had been granted bus access even though only one block actually has access granted from a bus arbiter.

30. A method according to claim 29, wherein the first block has bus access granted by the bus arbiter and wherein the second block uses internal logic to cause it to operate as if it has bus access.